Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 7656877, 11 pages
https://doi.org/10.1155/2021/7656877

Research Article

Antitamper Image Watermarking Based on Cellular Network Topology for IoT-Themed Mobile Forensics

Xiao-zhu Xie[1], **Ching-Chun Chang**[2,3], **Zhong-Liang Yang**[2] **and Li Li**[3]

[1]*School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China*
[2]*Department of Electronic Engineering, Tsinghua University, Beijing, China*
[3]*College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China*

Correspondence should be addressed to Ching-Chun Chang; ccc@fcu.edu.tw

Received 6 August 2021; Accepted 2 September 2021; Published 12 October 2021

Academic Editor: Chi-Hua Chen

The Internet of Things (IoT) connects physical and digital worlds with mobile devices, accompanied by a surge in cybersecurity issues. With the rapid adoption of mobile devices, mobile forensics emerges as a new interdisciplinary field that concerns many forms of sabotage and cybercrime in the context of mobile computing. One of the most common cyberattacks is tampering. Digital watermarking is a tamper-evident technique used to protect data integrity. In this paper, we present an antitamper image watermarking scheme designed for mobile communications with low computational cost. A reference matrix based on cellular network topology is introduced to guide the watermark embedding and extraction processes. This reference matrix serves as a lookup table to reduce computational complexity, thereby enabling efficient implementation on mobile devices. Our scheme is aimed at offering high accuracy in detecting and localizing tampered regions. We also achieve a high watermarking capacity while leaving the visual quality of the carrier images nearly unharmed. Experimental results validate the effectiveness of our scheme against various types of simulated forgery including cropping and copy/paste attacks.

1. Introduction

The Internet of Things (IoT) bridges the gap between the physical and digital worlds and, at the same time, brings new cybersecurity challenges [1]. One of the most common cyberattacks is data tampering, which may take place during transmission across the Internet. Data integrity is crucial to many applications in IoT, and hence, it is important to develop an authentication scheme compatible with the IoT environment. Watermarking is an established technique used to protect copyright and authenticate the integrity of images [2, 3]. It embeds the secret information (i.e., Watermark) into the protected image invisibly. On the receiver side, the watermark is extracted and used for the proof of ownership or the authentication. Watermarking schemes can be broadly categorized as either robust watermarking schemes, typically designed to protect copyright, or fragile watermarking schemes, where the aim is to protect integrity. Furthermore, attack resisting and tampering localization are additional metrics used to evaluate the performance of watermarking schemes.

Since the embedding of watermark inevitably decreases the image quality, a trade-off must be made between the marked image quality and the ability to perform attack resisting and tampering localization. We propose an image tampering detection scheme based on blind and fragile watermarking scheme that maintains good image quality. The main contributions of our scheme can be summarized as below:

(i) This paper designs a cellular network reference matrix, based on which a blind and fragile watermarking scheme is proposed

(ii) The proposed scheme achieves a high embedding capacity of 1.403 bpp and maintains a good image quality of 47.16 dB

(iii) The proposed scheme can effectively resist attacks including cropping attack and copy and paste

attack, and it provides high accuracy tampering localization with tampering detection rate (TDR) of 100% after one dilation operation for cropping attacks

The structure of this paper is organized as follows. Section 2 provides a literature review. Section 3 introduces a preliminary work by which our cellular network matrix is inspired. Section 4 presents the proposed scheme, followed by the experimental results and analysis in Section 5. Finally, conclusions are drawn in Section 6.

2. Related Work

In robust watermarking schemes, watermark information, e.g., logos or copyright information, is embedded into the host image in an imperceptible way. The receiver can extract the watermark with accuracy even if the marked image has been tampered with. In other words, the verification of copyright can be guaranteed even if the marked image is subjected to malicious attacks. Since the frequency domain is more robust than the spatial domain against most attacks, robust watermarking schemes are mainly conducted in the frequency domain. In [4], the authors provide a robust watermarking scheme based on integer wavelet transform (IWT). It embeds the hash value of ROI (region of interest) into ROIN (region of noninterest) using IWT. Though it performs well on verifying the integrity of ROI, it is only applicable for images with small region of interest. Liu et al. [5] proposed a robust watermarking scheme by embedding the watermark into the LL subband after discrete wavelet transform (DWT) decomposition. Since the LL subband would hardly be changed when suffering the malicious attacks, it has strong robustness. However, it does not provide the tampering localization. Abdulrahman and Ozturk [6] applied discrete cosine transform (DCT) and DWT to each color component of an RGB image, then embed the watermark into four DWT bands. This approach is resistant to linear and nonlinear attacks. Singh and Bhatnagar [7] first transform the host image into the integer DCT domain before dividing it into nonoverlapping blocks in which the watermark is embedded using singular value decomposition. The scheme in [7] is robust against not only common attacks but also geometric attacks. However, the computational complexity of [7] is relatively high and not suitable for mobile computing environment. Yi et al. [8] generated a binary watermark by conducting double random-phase encoding on the host image itself and then embedded it into the DCT coefficients of the protected image. This scheme provides a robust authentication technique that can resist noise attacks, filtering attacks, partial occlusion attacks, etc. Yet, it fails to accomplish authentication if the error rate of the second phase key exceeds 20%. Robust watermarking schemes can protect against a wide range of attacks; however, image quality is relatively low, and the tampered region cannot be located.

Fragile watermarking schemes are typically designed for integrity authentication. Any slight modification to the marked image can be detected sensitively, and the tampered region can be reliably located. In this light, most existing schemes are based in the spatial domain. In early schemes, a watermark is generated with a secret key and a designated function. At the verifier side, the embedded watermark is extracted and compared with the original watermark. If there exists any difference, it indicates that the marked image has been tampered with. For instance, Yeung and Mintzer [2] use a secret key to generate a binary valued function which ensures that the extracted watermark will be the same as the embedded one if there exists no tampering. However, the scheme in [2] cannot specify the tampered region. Later, some block-based fragile watermarking schemes with tampering localization are proposed. Qin et al. [9] embed the authentication bits into the least significant bit (LSB) of the central pixel in one block and propose a block-wise fragile watermarking scheme with tampering localization. It provides tampering localization, at the cost of the image quality. Zhang et al. [10] generate a binary watermark by performing a local binary pattern (LBP) operation on nonoverlapping blocks of the original image, then embed the watermark using LSB substitution. Similar to [10], Gul and Ozturk [11] generate a watermark by performing the SHA-256 hash function on some divided blocks, then embedding the watermark in the LSBs of other blocks. Bhalerao et al. [12] propose a watermark using the secure hashing algorithm SHA-1 hash function. Though the methods in [8–12] can provide tampering localization and resist various attacks, they are based on blocks, which means they detect and locate the tampering region in the unit size of a block, resulting in a low resolution.

To improve the accuracy of tampering detection, some pixel-wise watermarking schemes are proposed. Prasad and Pal [13] take two pixels as a unit, then compute the watermark from the two most significant bits (MSBs) of each pixel using Hamming code, and finally embed them in the LSBs of the same pixels. Gong et al. [14] firstly generate two watermarks, i.e., the diffusion watermark and the authentication watermark, then arbitrarily embed them into the two LSBs of the cover image. Both watermarks are sensitive to alteration of the cover image, and the authentication watermark provides tampering localization at pixel level. Similarly, Memon and Alzahrani [15] generate a fragile watermark and a robust watermark, then, respectively, embed them into the region of interest (ROI) and the region of noninterest (RONI) of CT scan images. These schemes give good performance on tampering detection and localization; however, the image quality of the marked image is low.

3. Preliminary Work

Chang et al. [16] proposed the turtle shell matrix-based data hiding scheme in 2014. They first put forward the concept of turtle shell matrix, which is sized 256×256 and composed of turtle shells. The turtle shell is defined as a hexagon containing eight distinct digits, including two digits inside the hexagon, and six digits on the edges. The construction rules are as follows: (1) select a number from 0 to 7 to initialize the element with the coordinate of (0, 0); (2) the values of elements in the same row increase in steps of "1" modulo 7, and the values of elements in the same column increase in alternate steps of "2" and "3" modulo 7. An example of turtle shell matrix can be seen in Figure 1.

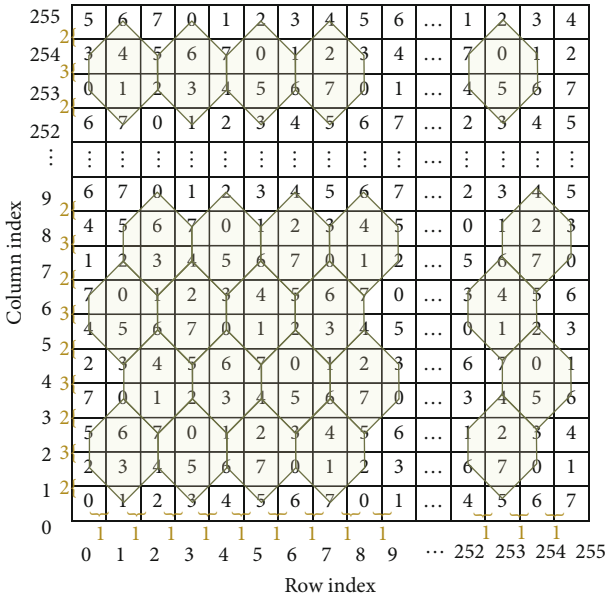


Figure 1: An example of turtle shell matrix.

Looking into the matrix, we can see that for any element, eight distinct digits ranging from 0 to 7 can be found nearby. In this light, Chang et al. regarded one pixel pair as the coordinate of the matrix and embedded a base-8 digit into one pixel pair. Interested readers can refer to [16] for more details about finding the closest element whose value equals to the to-be-embedded digit.

4. Proposed Scheme

We design a cellular network matrix, with which we propose a blind and fragile watermarking scheme for image tampering detection. Figure 2 shows a flowchart of the proposed scheme. The watermark is secretly embedded into the original image using the cellular network matrix to generate the watermarked image. During transmission in a public channel, the watermarked image may be tampered with by a malicious attacker. On the receiver side, the tampered region can be detected using the cellular network matrix, and the watermark can be extracted. In the following sections, cellular network matrix construction, data embedding, data extraction, and tampering detection are described successively.

4.1. Cellular Network Matrix Construction. Inspired by the turtle shell matrix [17], we design the cellular network reference matrix $M = [m(i,j)]_{N\times N}$ $(i,j = 0, 1, 2,\cdots,N)$ according to the following rules, where N is determined by the bit depth of the cover image.

(1) Set the value of an element to "Null" when the sum of its row and column is even. For instance, $m(0,0)$, $m(1,1)$, $m(1,3)$, and $m(2,2)$ are "Null"

(2) Otherwise, the values of elements in the same row increase in steps of 1 modulo 7, and the values of elements in the same column increase in steps of 4 modulo 7

It should be noted that the values of elements with the coordinate of (1, 0) and (0, 1) should be initialized first. And the difference between two values is 5 modulo 7. For instance, if $m(1,0)$ is valued 1, then $m(0,1)$ gets the value of 6; if $m(1,0)$ is valued 4, then $m(0,1)$ gets the value of 2. Once the initial values are given, the matrix is determined. And the initial values are referred as auxiliary information which is shared with the data hider and the receiver for data embedding and extraction. Since one initial value is represented using three bits, the auxiliary information occupies six bits totally. The auxiliary information can be self-embedded into the host image using LSB (least significant bit) substitution in our algorithm, as the information is so little that it barely affects the performance of the proposed scheme. After initialization, the rules can be formulated as

$$m(i,j) = \begin{cases} \text{Null}, & \text{if } (i+j) \text{ is even}, \\ m(i-2,j)+1 \mod 7, & \text{if } i \geq 3 \text{ and } (i+j) \text{ is odd and } i \text{ is odd}, \\ m(i,j-2)+4 \mod 7, & \text{if } j \geq 3 \text{ and } (i+j) \text{ is odd and } j \text{ is odd}. \end{cases} \quad (1)$$

For easy description, we refer to elements containing null values as "null elements" and elements that contain valid values as "valid elements." Figure 3 shows two examples of the matrix. Looking into the generated cellular network matrix, two important features can be observed:

(1) The four direct neighbors of any valid element are null elements

(2) Any cell, drawn to include a central valid element, the four nearest neighbor valid elements, and the two valid elements immediately above and below, can be seen to include all seven unique digits from 0 to 6. Several examples of such cells are highlighted in Figure 3

4.2. Cellular Network Matrix-Based Data Hiding. According to the features of the cellular network matrix, we are able to use one cover image pixel pair to form the coordinates of a matrix element and hide one base-7 digit within them. It is noted that the proposed scheme is suitable for images of any bit depths as long as we extend the matrix size to adapt

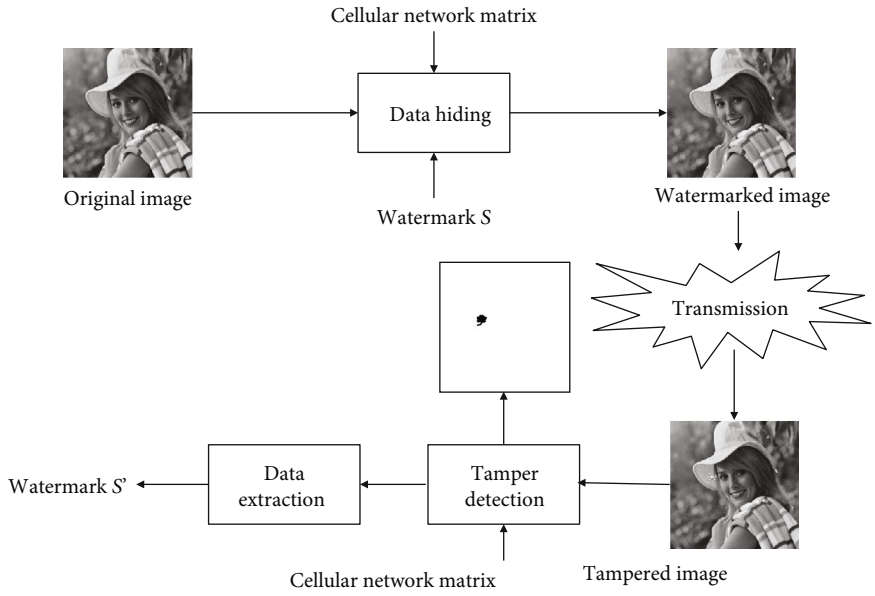


Figure 2: Flowchart of proposed scheme.

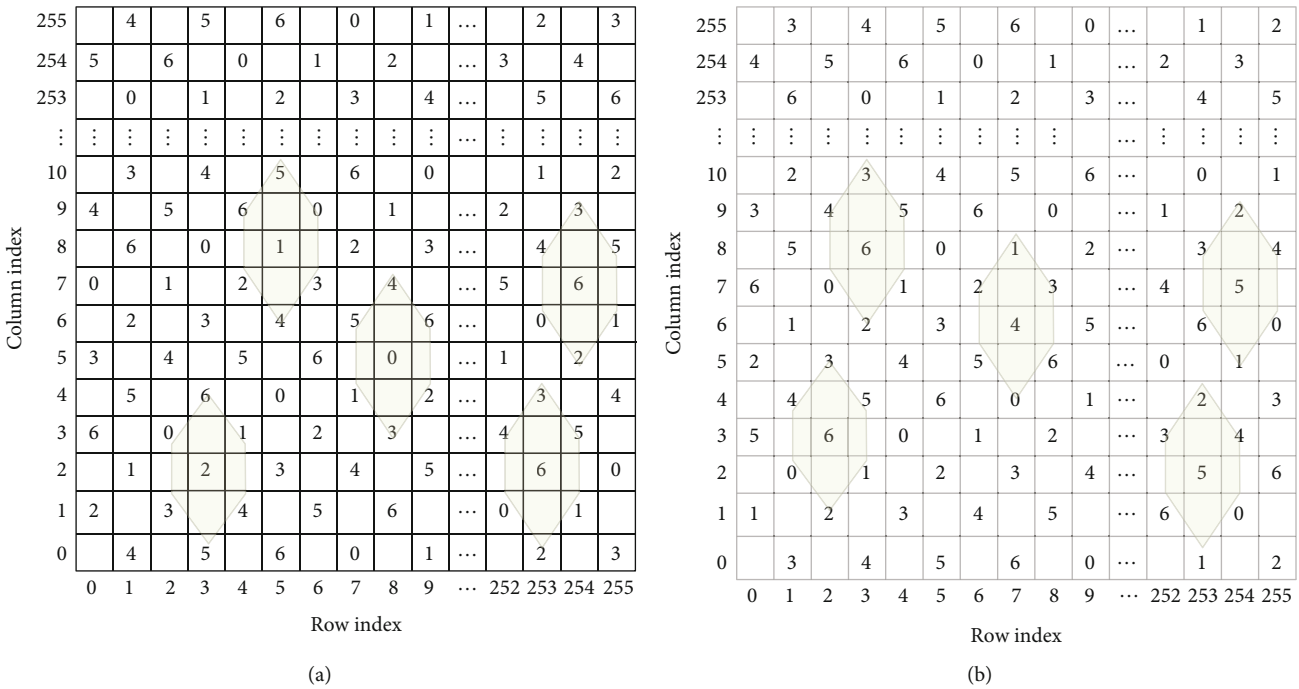


Figure 3: Examples of the cellular network matrix.

the bit depth. For simplicity, we take the gray image as the host image to depict the algorithm. The detailed processes are as follows:

(i) Inputs: the cover image, the binary watermark (generated by a random key), and the cellular network matrix

(ii) Output: a watermarked image

Step 1. Divide the cover image into nonoverlapping pixel pairs

Step 2. Convert the binary watermark into a base-7 digit stream

Step 3. Sequentially embed one digit w_i into each pixel pair (p_i, p_{i+1})

The guiding rule is to consider (p_i, p_{i+1}) as the coordinates of the cellular network matrix element and find the closest element (p'_i, p'_{i+1}) to (p_i, p_{i+1}) that satisfies

$$m\left(p'_i, p'_{i+1}\right) = w_i, \tag{2}$$

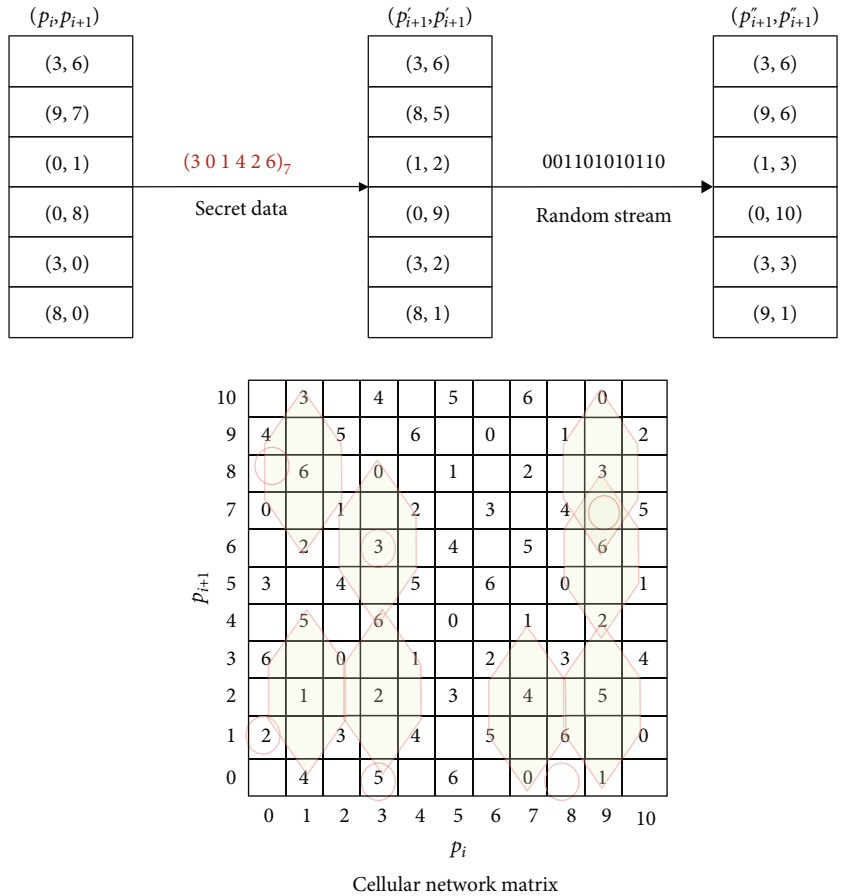


Figure 4: Examples of data hiding.

then replace (p_i, p_{i+1}) with (p'_i, p'_{i+1}) in the cover image. When the embedding is finished, we obtain the watermarked image.

The algorithm to find (p'_i, p'_{i+1}) satisfying Equation (2) can be categorized into four cases:

Case 1. $1 \le p_i \le 254$ and $2 \le p_{i+1} \le 253$ and $m(p_i, p_{i+1})$ is a valid element, for instance, (3,4). Then, draw a cell with center $m(p_i, p_{i+1})$ and find the element satisfying Equation (2) in the cell.

Case 2. $1 \le p_i \le 254$ and $2 \le p_{i+1} \le 253$ and $m(p_i, p_{i+1})$ is a null element, for instance, (5,5). Then, draw two cells with center $m(p_i, p_{i+1} - 1)$ and $m(p_i, p_{i+1} + 1)$, respectively. Note that one of the drawn cells may be not intact; it does not affect the result. Find the element satisfying Equation (2) in both cells.

Case 3. $m(p_i, p_{i+1})$ is a valid element and locates at the edge of the matrix, i.e., $p_i = 0$ or $p_i = 255$ or $p_{i+1} < 2$ or $p_{i+1} > 254$. Then, draw one cell using $m(p_i, p_{i+1})$ as the left bottom or the right bottom or the bottom or the top corner, respectively. Find the element satisfying Equation (2) in the cell.

Case 4. $m(p_i, p_{i+1})$ is a null element and locates at the edge of the matrix, i.e., $p_i = 0$ or $p_i = 255$ or $p_{i+1} < 2$ or $p_{i+1} > 254$. Then, draw one cell with corresponding center of $m(p_i + 1, p_{i+1})$, $m(p_i - 1, p_{i+1})$, $m(p_i, p_{i+1} + 1)$, or $m(p_i, p_{i+1} - 1)$, respectively. Find the element satisfying Equation (2) in the cell.

According to the embedding algorithm described above, when we divide the watermarked image into pixel pairs, the sum of these must be odd. In order to break this condition, which cannot be guaranteed for a natural image, we process the watermarked image using a pseudorandom binary stream according to

$$p'_i = p'_i + r_i, \tag{3}$$

where r_i is the elements of a pseudorandom binary stream $R = r_1, r_2, \cdots, r_{W*H}$ and W and H represent the width and height of the cover image, respectively.

To make the embedding algorithm easier to understand, we give examples in Figure 4. Assume that a string of base-7 digits, e.g., $(3\ 0\ 1\ 4\ 2\ 6)_7$, are embedded into cover pixel pairs {(3,6), (9,7), (0,1), (0,8), (3,0), (8,0)}. Pick up the first pixel pair (3,6) and the to-be-embedded digit "3," which belongs to Case 1 according to the embedding algorithm depicted above. Then, draw a cell in the matrix with center $m(3, 6)$, and find the element $m(3, 6)$ whose value equals "3" (see Figure 3). In this case, pixel pair (3,6) remains unchanged in the cover image. For the second pixel pair (9,7), which

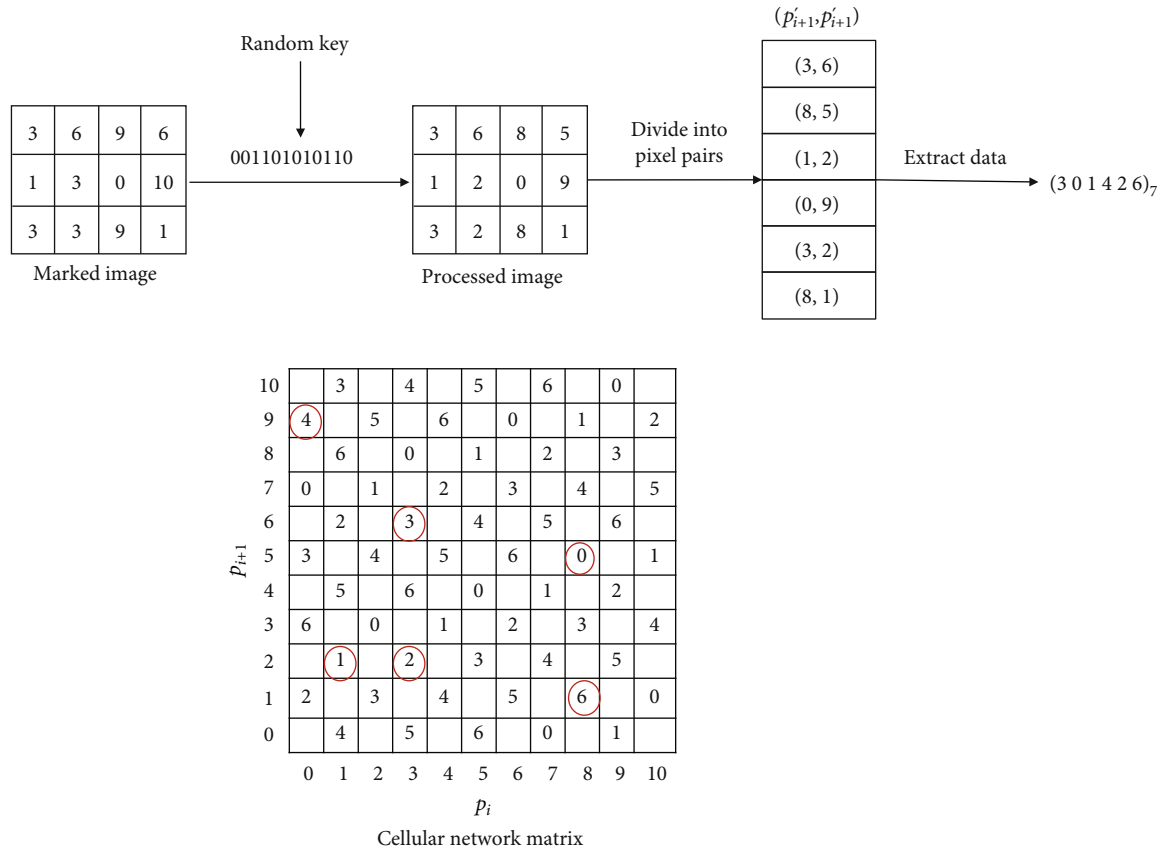


Figure 5: Example of data extraction.

belongs to Case 2, draw two cells with centers $m(9, 8)$ and $m(9, 6)$, respectively. Then, find the closest element whose value equals "0," $m(8, 5)$, and replace pixel pair (9,7) with pixel pair (8,5) in the cover image. The remaining four pixel-pairs belong to Cases 3 and 4. We do not describe these in the text due to the limited available space. After the six digits are embedded, we get the marked pixel pairs {(3,6), (8,5), (1,2), (0,9), (3,2), (8,1)}. In order to disguise the presence of the watermark, a pseudorandom binary stream, e.g., "001101010110" is added to the marked pixels, and the final marked pixel pairs are obtained, i.e., {(3,6), (9,6), (1,3), (0,10), (3,3), (9,1)}.

4.3. Data Extraction. The process of data extraction is the inverse operation of data embedding. The detailed procedures are as follows:

(i) Inputs: the marked image, the key, and the cellular network matrix

(ii) Output: the secret data

Step 1. Use the key to generate a binary stream $R = r_1, r_2, \cdots, r_{W*H}$ and process the watermarked image according to:

$$p'_i = p''_i - r_i, \tag{4}$$

where p'_i and p''_i represent the processed pixel and marked pixel, respectively

Step 2. Separate the processed image into pixel pairs, denoted as (p'_i, p'_{i+1})

Step 3. Consider each pixel pair as coordinates in the cellular network matrix and obtain the value $m(p'_i, p'_{i+1})$ which is precisely the secret digit encoded in base-7

Step 4. After all the secret digits are extracted, convert them into binary form which is the final secret data

An example is given in Figure 5 to illustrate the process of data extraction. As shown in Figure 5, a binary stream "001101010110" is generated using the key, then the processed image is obtained by operating Equation (4) on the marked image and the binary stream. After that, the processed image is separated into pixel pairs (p'_i, p'_{i+1}), and they are considered as coordinates of the cellular network matrix to get the value $m(p'_i, p'_{i+1})$. We now have the secret digits "301426" in base-7, and finally, we convert them into binary form to retrieve the secret data.

4.4. Tampering Detection Phase. If a marked image is tampered with during transmission, the recipient can detect the tampered region without knowing the original watermark. According to the embedding rule, if we select a pixel pair from the marked image and consider it as the coordinates of an element in the cellular network matrix, it must map to a "valid element." Therefore, we can determine that

a pixel pair has been tampered with if it maps to a "null element." Details of tampering detection are described as follows.

(i) Inputs: the marked image, the key, and the cellular network matrix

(ii) Output: the tampered region

Step 1. Follow the same procedures as depicted in steps 1-2 of data extraction

Step 2. Consider each pixel pair as the coordinates of an element in the cellular network matrix. If $m(p_i', p_{i+1}') = \text{Null}$, determine that at least one of the two pixels has been tampered with and marks the corresponding position in the tampered map as "1"; otherwise, mark it as "0"

Step 3. Since the tampered pixel pairs do not always map to the "null element," there is some misjudgment of the tampered map. Therefore, a dilation operation is performed on the interim tampered map to generate the final map. The operator is shown in

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{5}$$

5. Experimental Results

A series of experiments are conducted on the test images shown in Figure 6 to evaluate the performance of the proposed scheme. The test images are grayscale images with size 512×512 sampled from the UCID dataset [18]. Simulation experiments are conducted to evaluate the performance of tampering detection and localization when subjected to attacks, including cropping attack and copy and paste attack. Since we take only two pixels as a unit, tampering localization can be obtained with higher resolution than some state-of-the-art methods.

5.1. Evaluation Metrics. Definitions of the metrics used to evaluate the proposed scheme are described in this section.

(1) ER: embedding ratio

$$\text{ER} = \frac{N}{W \times H}\ (\text{bpp}), \tag{6}$$

where N represents the number of embedded secret bits and W and H denote the width and the height of the cover image, respectively.

(2) PSNR: peak-signal-to-noise-ratio, is calculated using

$$\text{PSNR} = 10 \log_{10}\left(\frac{255^2}{\text{MSE}}\right), \tag{7}$$

where MSE (mean square error) is computed as

$$\text{MSE} = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H}\left(I(i,j) - I'(i,j)\right)^2. \tag{8}$$

(3) SSIM: structural similarity index measure is calculated on various windows of an image. The measure between two windows x and y is

$$\text{SSIM}(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}, \tag{9}$$

where:

μ_x, μ_y are the average of x and y, respectively

σ_x^2, σ_y^2 are the variance of x and y, respectively

σ_{xy} is the covariance of x and y

c_1, c_2 are two variables to stabilize division with a weak denominator.

The larger the SSIM index is, the higher the similarity is. If SSIM = 1, then the two images are identical.

(4) Tampering detection rate (TDR) [19]: the percentage of tampered pixels that are correctly detected

$$\text{TDR} = \frac{\text{No.of correctly detected pixels}}{\text{No.of actual tampered pixels}} \times 100\%. \tag{10}$$

(5) False-positive rate (FPR) [19]: the percentage of nontampered pixels that are incorrectly judged to be tampered pixels

$$\text{FPR} = \frac{\text{No.of misjudged as tampered pixels}}{\text{No.of detected tampered pixels}} \times 100\%. \tag{11}$$

(6) False-negative rate (FNR) [19]: the percentage of tampered pixels that are incorrectly judged to be nontampered

$$\text{FNR} = \frac{\text{No.of misjudged as non} - \text{tampered pixels}}{\text{No.of detected tampered pixels}} \times 100\%. \tag{12}$$

5.2. Executing Efficiency. The execution times for seven test images with size of 512×512 are measured, and experiments are implemented using MATLAB 2017b on a PC with an Intel® Quad-Core i5 CPU @ 1.1GHz with 8 GB RAM. As shown in Table 1, matrix construction costs around 0.0027 s, the watermark embedding and extracting processes

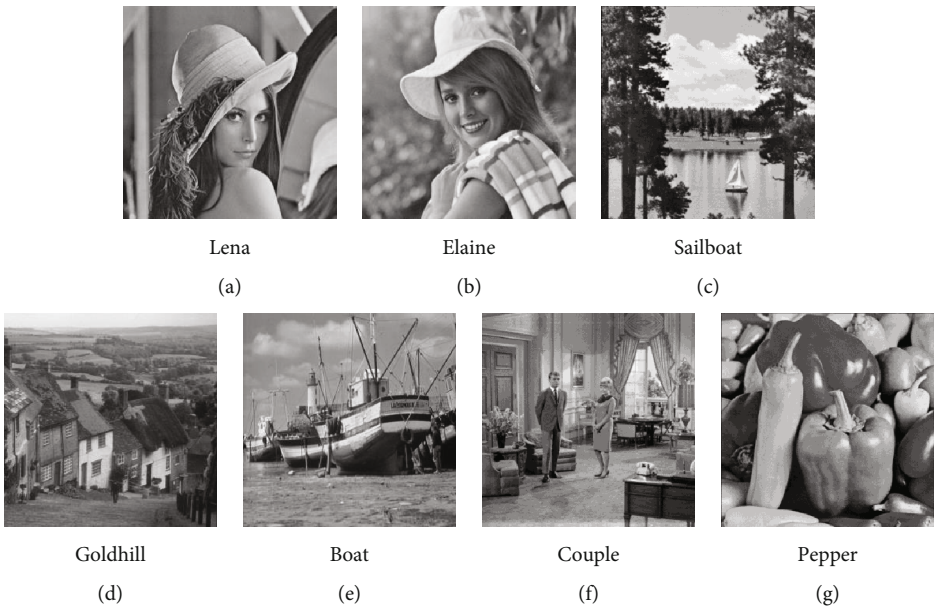


Figure 6: Seven test images.

Table 1: Execution times of the proposed scheme (unit: second).

Execution time	Matrix construction	Watermark embedding	Watermark extraction	Tamper detection	Total
Boat	0.0027	0.0226	0.0124	0.0028	0.0405
Couple	0.0027	0.0209	0.0120	0.0030	0.0386
Elaine	0.0027	0.0201	0.0132	0.0034	0.0394
Goldhill	0.0027	0.0197	0.0129	0.0029	0.0382
Lake	0.0027	0.0211	0.0156	0.0033	0.0427
Lena	0.0027	0.0258	0.0141	0.0036	0.0462
Pepper	0.0027	0.0205	0.0158	0.0029	0.0419
Average	0.0027	0.0215	0.0137	0.0031	0.0411

Table 2: Performance of the proposed scheme.

Image	ER (bpp)	PSNR (dB)	SSIM
Boat	1.403	47.17	0.9898
Couple	1.403	47.15	0.9938
Elaine	1.403	47.16	0.9925
Goldhill	1.403	47.16	0.9934
Lake	1.403	47.15	0.9929
Lena	1.403	47.17	0.9903
Pepper	1.403	47.16	0.9907
Average	1.403	47.16	0.9919

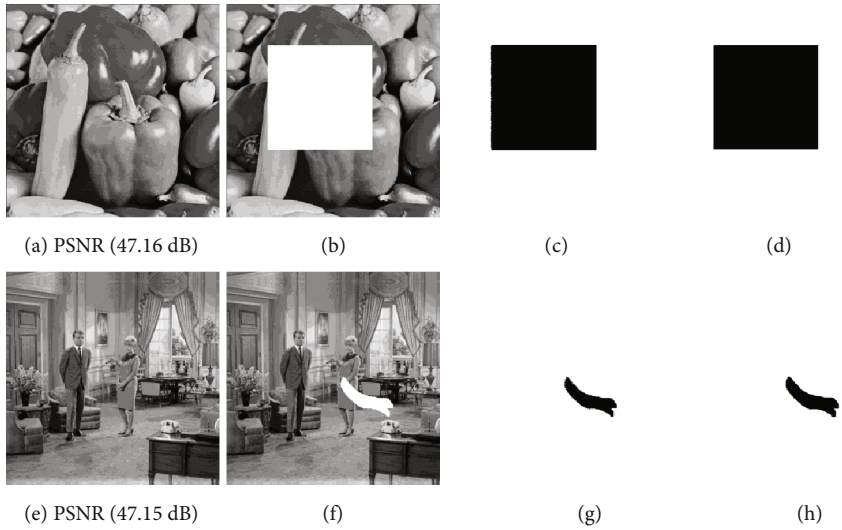
cost 0.0215 s and 0.0137 s, respectively, on average, and tampering detection costs around 0.0031 s. The total time cost is 0.0411 s on average. Therefore, we can declare that the proposed scheme has high efficiency, making it suitable for use in most real-time systems.

5.3. Image Quality and Embedded Data Capacity. In this section, three metrics, ER, PSNR, and SSIM, are used to evaluate the data capacity of the watermark and the imperceptibility of the watermark in watermarked images, with results summarized in Table 2. As described in Section 2.1, one base-7 digit is embedded into each pixel pair, giving a constant ER of 1.403 bpp. Corresponding PSNRs and SSIMs of the watermarked images with maximum payload are given as well. We can see that the average PSNR reaches 47.16 dB and an average SSIM of 0.9919 is achieved, which implies that very good imperceptibility of the watermarked is achieved by the proposed scheme.

5.4. Tampering Detection and Localization

5.4.1. Cropping Attack Detection. First, simulation experiments of cropping attack are conducted on the watermarked images "Pepper" and "Couple," then the proposed scheme is used to detect and locate the tampered region, with the results shown in Figure 7.

Figures 7(a) and 7(e) show the watermarked images, and Figures 7(b) and 7(f) show the tampered images with regular and irregular cropping, respectively. In more detail, "Pepper" is tampered with by cropping a rectangular region, and "Couple" is tampered with by cropping an irregular shape. Since all the tampered pixel values become 255, the sum of two adjacent pixels is even. In other words, pixel pairs from the tampered region inevitably map to a "null



(a) PSNR (47.16 dB) (b) (c) (d)

(e) PSNR (47.15 dB) (f) (g) (h)

Figure 7: Tampering localization of cropping attack: (a), (b), (c), (d) for "Pepper"; (e), (f), (g), (h) for "Couple."

Table 3: Detection results of cropping attack.

Cover images	Results without dilation operation (%)			Results with one dilation operation (%)		
	TDR	FPR	FNR	TDR	FPR	FNR
Peppers	99.80	0.20	0.06	100	1.70	0
Couples	98.78	1.41	0.02	100	8.80	0

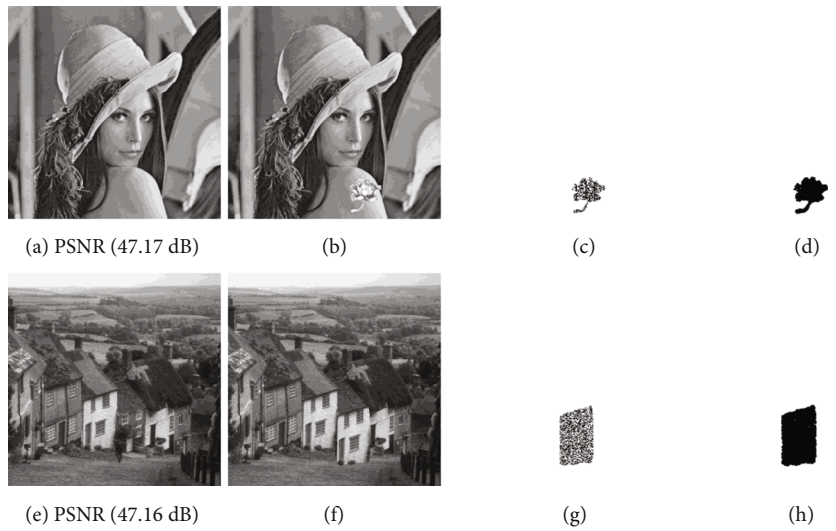
element" in the matrix and are detected as tampered pixels. Therefore, the tampered region can be detected accurately except for those pixel pairs on the tampering edge. The detection results without the dilation operation are shown in Figures 7(c) and 7(g), with TDRs of 99.80% and 98.78%, respectively. After one dilation operation, both TDRs are improved to 100% (as shown in Figures 7(d) and 7(h)). Without doubt, the dilation operation would make some nontampered pixels be tampered ones, resulting in an increase of FPR. The statistical results can be found in Table 3, from which we can see that after one dilation operation, the TDR reaches up to 100% with an FNR of 0%, yet the FPR increases slightly. Trade-off between TDR and FPR can be made by conducting the dilation operation.

5.4.2. Copy and Paste Attack Detection. Next, simulation experiments of copy and paste attack are conducted on two watermarked images "Lena" and "Goldhill" generated by the proposed scheme (as shown in Figures 8(a) and 8(e)). Copy and paste attack refers to an attack that replaces one region in the watermarked image with a copy of another region either from within the same image or from an external image. For instance, Figure 8(b) shows an image tampered with by copying a flower onto the shoulder of "Lena" and Figure 8(f) shows an image tampered with by copying a section of white wall and pasting it back on top of a different region within the same image. The detection results without dilation operation can be seen in Figures 8(c) and 8(g), whose TDRs are 55.44% and 49.15%, respectively. FPRs and FNRs are relatively low as well. The statistical results can be found in Table 4, from which we can see that after one dilation operation, TDRs increase shapely; meanwhile, FNRs decrease considerably. Regrettably, FPRs increase accordingly. After two dilation operations, the detection performance improves remarkably (as shown in Figures 8(d) and 8(h)). Both TDRs are close to 100%, and the FNRs are close to 0%. However, the FPRs increase to 16.58% and 7.36%, respectively. The trade-off can be made by the number of dilation operations applied.

5.5. Comparison. Comparisons with some relevant schemes [8, 11, 13, 14] are made in this section. Among them, the scheme in [8] is conducted in the DCT domain, while the remaining schemes are conducted in the spatial domain. Table 5 gives the comparison results. We define unit size to be the dimensions in pixels of the region used for tampering detection and localization. The smaller the unit size, the higher the resolution for detection and localization. We can see that the proposed scheme is significantly superior to schemes [8, 11] in terms of resolution and embedded data capacity. The resolution of the proposed scheme is the same as scheme [13], and it has a similar embedding ratio; however, it performs significantly better in terms of image quality and TDR. Though scheme [14] has a better resolution and a larger embedding rate, the proposed scheme again performs significantly better in terms of image quality and TDR. Overall, the proposed scheme performs well in terms of embedding capacity and tampering detection ability compared to most existing methods and gives significantly improved image quality.

6. Conclusions

We propose a high-performance blind fragile watermarking scheme for image tampering detection. The core contribution of this work is the construction of a cellular network



(a) PSNR (47.17 dB) (b) (c) (d)

(e) PSNR (47.16 dB) (f) (g) (h)

Figure 8: Tampering localization of copy and paste attack: (a), (b), (c), (d) for "Lena"; (e), (f), (g), (h) for "Goldhill."

Table 4: Detection results of copy and paste attacks.

Cover images	Results without dilation operation (%)			Results with one dilation operation (%)			Results with two dilation operations (%)		
	TDR	FPR	FNR	TDR	FPR	FNR	TDR	FPR	FNR
Lena	55.53	3.08	0.57	94.24	9.47	0.08	99.58	16.58	0.01
Goldhill	49.15	2.18	2.04	93.54	4.13	0.26	99.79	7.36	0.01

Table 5: Comparisons between the proposed scheme and others.

Scheme	Domain	Unit size	ER	PSNR (dB)	TDR
[8]	DCT	8×8	4096 bits	42.23	-
[11]	Spatial	32×32	0.25 bpp	≥57	-
[13]	Spatial	1×2	1.5 bpp	42.09	99.50%
[14]	Spatial	1×1	2.0 bpp	44.17	99%
Ours	Spatial	1×2	1.403 bpp	47.16	100%

"-": not present.

reference matrix which is central to a suite of watermarking algorithms that we implement, evaluate, and compare with existing results. Experimental results show that the proposed scheme can achieve a high embedding capacity of 1.403 bpp while maintaining excellent image quality with peak signal-to-noise ratio of 47.16 dB. If the watermarked image is tampered with during transmission, the tampering region can be detected and located with high resolution without knowing the original watermark. For cropping attack and copy and paste attack, the proposed cellular network scheme can provide highly accurate tampering detection, obtaining a tampering detection rate of up to 100%. Furthermore, the algorithm is easy to realize, has high efficiency, and is suitable for use in mobile computing environment.

As the increasing concerns of information integrity, reversible watermarking techniques gain more and more concerns. The technique realizes that the watermarked image can be recovered thoroughly after the watermark is extracted. Chang et al. [20–22] propose several reversible watermarking schemes for protecting privacy in the cloud computing environment. Thus, in the future, we will do research on reversible watermarking for image authentication and tampering localization.

Data Availability

The image data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 62172132) and Scientific Research Foundation for the Introduction of Talent at Xiamen University of Technology (YKJ21007R).

References

[1] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[2] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," *Proceedings of international conference on image processing*, vol. 2, pp. 680–683, 1997.

[3] A. K. Kamran and S. A. Malik, "A high capacity reversible watermarking approach for authenticating images: exploiting down-sampling, histogram processing, and block selection," *Information Sciences*, vol. 256, pp. 162–183, 2014.

[4] R. Eswaraiah and E. Sreenivasa Reddy, "Robust medical image watermarking technique for accurate detection of tampers inside region of interest and recovering original region of interest," *IET Image Processing*, vol. 9, no. 8, pp. 615–625, 2015.

[5] X.-L. Liu, C.-C. Lin, and S.-M. Yuan, "Blind dual watermarking for color images' authentication and copyright protection," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 28, no. 5, pp. 1047–1055, 2018.

[6] A. K. Abdulrahman and S. Ozturk, "A novel hybrid DCT and DWT based robust watermarking algorithm for color images," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 17027–17049, 2019.

[7] S. P. Singh and G. Bhatnagar, "A new robust watermarking system in integer DCT domain," *Journal of Visual Communication and Image Representation*, vol. 53, pp. 86–101, 2018.

[8] F. Yi, Y. Kim, and I. Moon, "Secure image-authentication schemes with hidden double random-phase encoding," *IEEE Access*, vol. 6, no. 1, pp. 70113–70121, 2018.

[9] C. Qin, P. Ji, X. Zhang, J. Dong, and J. Wang, "Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy," *Signal Processing*, vol. 138, pp. 280–293, 2017.

[10] H. Zhang, C. Wang, and X. Zhou, "Fragile watermarking based on LBP for blind tamper detection in images," *Journal of Information Processing Systems*, vol. 13, no. 2, pp. 385–399, 2017.

[11] E. Gul and S. Ozturk, "A novel hash function based fragile watermarking method for image integrity," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 17701–17718, 2019.

[12] S. Bhalerao, I. A. Ansari, and A. Kumar, "A secure image watermarking for tamper detection and localization," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1057–1068, 2021.

[13] S. Prasad and A. K. Pal, "A tamper detection suitable fragile watermarking scheme based on novel payload embedding strategy," *Multimedia Tools and Applications*, vol. 79, no. 3-4, pp. 1673–1705, 2020.

[14] X. Gong, L. Chen, F. Yu, X. Zhao, and S. Wang, "A secure image authentication scheme based on dual fragile watermark," *Multimedia Tools and Applications*, vol. 79, no. 25-26, pp. 18071–18088, 2020.

[15] N. Memon and A. Alzahrani, "Prediction-based reversible watermarking of CT scan images for content authentication and copyright protection," *IEEE Access*, vol. 8, pp. 75448–75462, 2020.

[16] C.-C. Chang, Y. Liu, and T. S. Nguyen, "A novel turtle shell based scheme for data hiding," in *Proceedings of 2014 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 89–93, Kita Kyushu, Japan, 2014.

[17] X. Z. Xie, C. C. Chang, C. C. Lin, and J. L. Lin, "A turtle shell based RDH scheme with two-dimensional histogram shifting," *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 19413–19436, 2019.

[18] *USC-SIPI Image Database*http://sipi.usc.edu/database/.

[19] G. D. Su, C. C. Chang, and C. C. Lin, "High-precision authentication scheme based on matrix encoding for AMBTC-compressed images," *Symmetry*, vol. 11, no. 8, p. 996, 2019.

[20] C.-C. Chang, C.-T. Li, and Y. Q. Shi, "Privacy-aware reversible watermarking in cloud computing environments," *IEEE Access*, vol. 6, pp. 70720–70733, 2018.

[21] C.-C. Chang, C.-T. Li, and K. Chen, "Privacy-preserving reversible information hiding based on arithmetic of quadratic residues," *IEEE Access*, vol. 7, pp. 54117–54132, 2019.

[22] C.-C. Chang, "Neural reversible steganography with long short-term memory," *Security and Communication Networks*, vol. 2021, Article ID 5580272, 14 pages, 2021.